

- 1) Are we supposed to be able to solve deep differentiation of Neural networks by hand? (i.e.):

**Exercise 1. Derive BackProp**

A network with 2 hidden layers followed by an output layer has the output in the final layer (=3rd layer)

$$\hat{y}_n^\mu = x_n^{(3)} \quad (1)$$

$$= g^{(3)}[a_n^{(3)}] \quad (2)$$

$$= g^{(3)}\left[\sum_i w_{ni}^{(3)} g^{(2)}[a_i^{(2)}]\right] \quad (3)$$

$$= g^{(3)}\left[\sum_i w_{ni}^{(3)} \left[g^{(2)}\left[\sum_j w_{ij}^{(2)} x_j^{(1)}\right]\right]\right] \quad (4)$$

$$= g^{(3)}\left[\sum_i w_{ni}^{(3)} \left[g^{(2)}\left[\sum_j w_{ij}^{(2)} g^{(1)}(a_j^{(1)})\right]\right]\right] \quad (5)$$

$$= g^{(3)}\left[\sum_i w_{ni}^{(3)} \left[g^{(2)}\left[\sum_j w_{ij}^{(2)} g^{(1)}\left(\sum_k w_{jk}^{(1)} x_k^\mu\right)\right]\right]\right] \quad (6)$$

We use an error function

$$E(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)}) = \frac{1}{2} \sum_\mu \sum_n [t_n^\mu - \hat{y}_n^\mu]^2 \quad (7)$$

Calculate the derivative  $\partial E / \partial w_{52}^{(1)}$ , hence the derivative with respect to the weight connecting the second input to the fifth hidden neuron in the first layer.

i.

- 2) Do we need to know biology stuff ( Pre-synaptic and post-synaptic factor)?
- 3) Do we need to do any form of coding on the exam?
- 4) What does bootstrapping mean in the context of ANN ( Is it simply using random variables in the network and updating them until we reach some sort of convergence, or is it the weird thing from statistics [https://en.wikipedia.org/wiki/Bootstrapping\\_\(statistics\)](https://en.wikipedia.org/wiki/Bootstrapping_(statistics)) , or is it something else?)
- 5) Eligibility traces ( I don't know enough to know what to ask here, so you might skip this one)
- 6) Why is it ok that we rewrite  $\nabla_\theta \int_x p(x; \theta) * R(x) dx$  to  $\nabla_\theta \int_x \pi(x; \theta) * R(x) dx$  when we find the online update rule in reinforcement learning? ( This might have been answered on the forums, I did not understand it, but I can probably try to go back and see if I can figure it out)
- 7) How does the Backprop-differentiation algorithm work (I might figure this one out when I go through the slides, though)
- 8) If there are online-rules that can handle gradient descent, are there any mathematical reasons for using mini-batches in the training of networks, or are the reasonings mostly empirical? (I guess the answer might just be "Because the minibatch has less variance"...) )
- 9) Do we need to know ADAM, and how it differs from SGD?
- 10) Will the exam tend more towards:
  - a. Text based questions, where we have to know and reason about different things that have been explained about neural networks
  - b. Being able to understand the math behind Neural Networks sufficiently well to be able to deduce other new concepts if we are asked to do so during the exam?
  - c. Knowing the material well enough by heart to be able to solve a lot of tasks very quickly?
- 11) Vanishing gradient problem. ( This one might just be in the slides/exercises, so it might be fine)